OPEN ACCESS

University of BRISTOL

Song, H., Diethe, T., Kull, M., & Flach, P. (2019). Distribution Calibration for Regression. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *International Conference on Machine Learning, 9-15 June 2019, Long Beach, California, USA* (pp. 5897-5906). (Proceedings of Machine Learning Research; Vol. 97). http://proceedings.mlr.press/v97/song19a.html

Peer reviewed version

Link to publication record in Explore Bristol Research

PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via PMLR at http://proceedings.mlr.press/v97/song19a.html . Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/

arXiv:1905.06023v1 [stat.ML] 15 May 2019

Distribution Calibration for Regression

Hao Song [1] **Tom Diethe** [2] **Meelis Kull** [3] **Peter Flach** [1 4]

Abstract

We are concerned with obtaining well-calibrated output distributions from regression models. Such distributions allow us to quantify the uncertainty that the model has regarding the predicted target value. We introduce the novel concept of distribution calibration, and demonstrate its advantages over the existing definition of quantile calibration. We further propose a *post-hoc* approach to improving the predictions from previously trained regression models, using multi-output Gaussian Processes with a novel Beta link function. The proposed method is experimentally verified on a set of common regression models and shows improvements for both distribution-level and quantile-level calibration.

1 Introduction

With recent progress in predictive machine learning, many models are now capable of providing outstanding performance with respect to certain metrics, such as accuracy in classification or mean squared error in regression. While such models are suitable for some tasks, they often cannot provide well-quantified uncertainties on the target variables. In this paper we focus on this problem in the regression setting, extending concepts from the well-established framework of probability calibration for classification.

In a classification task, a probabilistic prediction $s \in [0, 1]$ for the positive class is calibrated if the following condition holds: among all the instances receiving this same prediction value s, the probability of observing a positive label is s. Having such calibrated outputs is important as they can be interpreted as degrees of uncertainty on the class, hence enabling quantitative approaches towards decision making, such as cost-sensitive classification [33]. However, from simple models (*e.g.* Naïve Bayes) to complex ones (*e.g.* (deep) Neural Networks (NNs)), poor calibration is often observed, irrespective of the model's complexity or its probabilistic nature [9, 16]. To mitigate this, several techniques have been proposed to apply *post-hoc* corrections to the outputs from trained classifiers such as Platt scaling [23], Isotonic regression [33], and Beta calibration [16].

In the setting of regression, calibration has been traditionally defined through predicted credible intervals [7, 3, 14], where a 0.95 predicted credible level (e.g. conditional quantile) is calibrated if marginally 95% of the true target values are below or equal to it. Having a quantile-calibrated regressor is particularly useful for certain forecasting tasks such as energy usage [7] and supply chain optimisation [12]. Existing approaches which aim to obtain calibrated predictions as part of the training, can be loosely divided into two categories: (i) quantile regression [13], where it has shown that generalised additive models can be applied to yield better-calibrated quantiles [3]; (ii) direct application of conditional density estimators [2, 30, 22] to obtain an estimated cumulative distribution function (CDF), which can then be used to generate corresponding credible intervals.

While such approaches can be good options when simple models will suffice, they are less suitable when employing (possibly extant) specialised models, such as (pre-trained) deep NN models. Unlike the case of classification, the *post-hoc* calibration approach in regression has been left largely unexplored. Recently, [14] proposed a *post-hoc* approach that applies isotonic regression to match the predicted CDF and empirical frequency, so that the final results are better calibrated in the quantile sense.

While quantile-level calibration is useful in certain scenarios, it is defined uniformly over the entire input space and does not ensure calibration for a particular prediction, unlike the classification setting. For instance, a quantile-calibrated regressor cannot always guarantee that all instances receiving an estimated mean of μ and standard deviation σ are indeed distributed as a Gaussian distribution with the same moments. In this paper, we focus on a *post-hoc* method that aims to achieve distribution-level calibration and hence gives more accurate uncertainty information for a continuous target variable.

[1]University of Bristol, Bristol, United Kingdom [2]Amazon Research, Cambridge, United Kingdom [3]University of Tartu, Tartu, Estonia [4]The Alan Turing Institute, London, United Kingdom. Correspondence to: Hao Song <hao.song@bristol.ac.uk>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

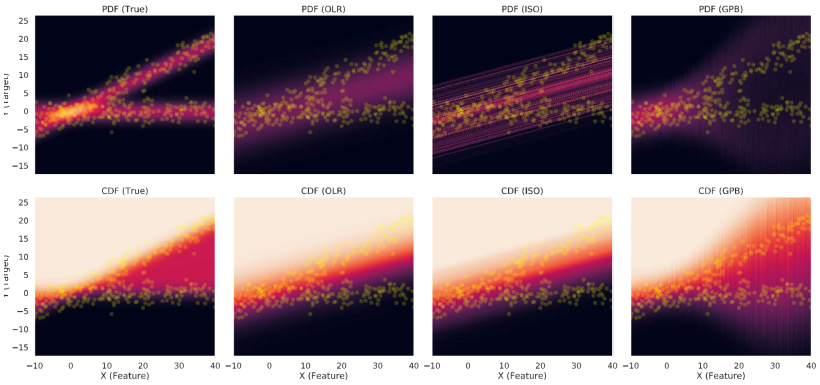

Figure 1. Applying quantile and distribution calibration on a synthetic dataset. The column on the left shows the true conditional PDF / CDF of the dataset, with the yellow points being the observed data. An ordinary linear regression is fitted to the data and the predictions are shown in the second column. An isotonic regression and a GP-BETA model are trained to calibrate the OLS outputs on quantile and distribution level respectively, giving the right two columns.

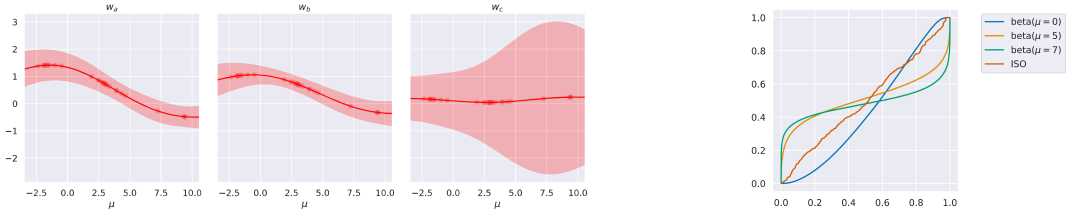

Figure 2. (**Left Three**) The latent Beta parameters modelled by the GP, with red points representing the inducing points. The red shade shows the area for one standard deviation at each side. Notice here the x-axis represent the mean value predicted by the OLS, and hence corresponds to the y-axis in Fig. 1. (**Right**) Calibration maps from GP-BETA and isotonic regression. GP-BETA is capable of predicting different calibration maps for different model output, the example shows three calibration maps at 0, 5, 7 respectively. Isotonic regression only gives a single calibration map for all model outputs, and it only aims to calibrate the marginal quantiles.

Our contributions are as follows:

1. We introduce the concept of distribution calibration, and demonstrate that being calibrated on a distribution-level will naturally lead to calibrated quantiles.
2. We propose a multi-output Gaussian Process (GP)-based approach [1] to solve the task of *post-hoc* density calibration. This approach models the distribution over calibration parameters, and uses a novel Beta link function to calibrate any given regression outputs. An example is demonstrated in Fig. 2.
3. To ensure the scalability of the model, we further provide a solution based on stochastic variational inference together with induced pseudo-points.
4. Finally, the proposed approach is experimentally analysed on different regression models including GP regression [24] and Bayesian NN (BNN) regression [5].

The rest of the paper is organised as follows. In Sec. 2 we introduce calibration in the context of both classification and regression, together with some related *post-hoc* calibration approaches. Sec. 3 defines density-level calibration and discusses some theoretical properties. The proposed calibration approach is described in Sec. 4. Experimental analysis is shown in Sec. 5 and Sec. 6 concludes.

2 Background and Definitions

Throughout this paper, X and Y are random variables over spaces $\mathbb{X}$ and $\mathbb{Y}$, where X represents the input features of an instance and Y is the corresponding target value. In k-class classification Y is categorical with $\mathbb{Y} = \{1, \ldots, k\}$.

Post-hoc calibration applies if there is a (pre-trained) probabilistic model, which inputs the feature values and outputs a probability distribution over the target value. We use the notation $\mathsf{f} : \mathbb{X} \to \mathbb{S}_\mathbb{Y}$ to denote such a probabilistic model, where $\mathbb{S}_\mathbb{Y}$ is a space of probability distributions over $\mathbb{Y}$. For the classification task, $\mathbb{S}_\mathbb{Y}$ consists in vectors $\mathbf{s} = [s_1, \ldots, s_k]$, where s_i denotes the probability of class i. Hence, $s_1, \ldots, s_k \in [0, 1]$ and $\sum_{j=1}^{k} s_i = 1$.

The idea of *post-hoc* calibration is to learn a transformation, which takes in the probability distribution as output by the model, and transforms it so that the resulting probability distribution would be better calibrated. Intuitively, being calibrated means some kind of an agreement between predicted distribution and actual empirical distribution. Next,

we will see how existing work has instantiated this intuitive notion for classification and regression, and propose a new notion of being distribution-calibrated for regressors.

2.1 Calibration in Classification

Classification calibration is defined as follows [15]:

Definition 1 (Calibrated Classifier)**.** *Assume we have a pair of jointly distributed random variables* (X, Y) *over* $\mathbb{X}$ *and* $\mathbb{Y} = \{1, \ldots, k\}$*, and a model* $\mathsf{f} : \mathbb{X} \to \mathbb{S}_{\mathbb{Y}}$*. Denoting* $S = \mathsf{f}(X)$ *as the random vector of predicted class probabilities,* f *is said to be calibrated iff* $\forall \mathbf{s} = [s_1, \ldots, s_k] \in \mathbb{S}_{\mathbb{Y}}$*, the following holds:*

$$\mathsf{P}(Y = j \mid S = \mathbf{s}) = s_j \,. \tag{1}$$

Alternative definitions exist, e.g. [9] require the accuracy on all instances with the same confidence level (highest predicted probability across classes) to agree with the confidence value.

As mentioned above, some models tend to give uncalibrated outputs due to certain computational heuristics or unrealistic assumptions. Some *post-hoc* approaches are hence proposed to adjust such outputs to yield better calibrated probabilities. Broadly speaking, most *post-hoc* approaches can be formalised as a calibration map $\mathsf{c} : \mathbb{S}_{\mathbb{Y}} \to \mathbb{S}_{\mathbb{Y}}$. In the binary classification case, a calibration map can be seen as a function $\mathsf{c} : [0, 1] \to [0, 1]$, transforming the probability of class 1, as the class 2 is simply the complement. The calibration map can then be visualised in a unit square as in Fig. 2 (Right). We hence proceed by introducing two illustrated calibration approaches for binary classification.

Isotonic calibration is a powerful non-parametric method based on isotonic regression along with a simple iterative algorithm called Pool Adjacent Violators (PAV), which finds the train-optimal regression line (calibration map) among all non-decreasing functions [33, 4]. The method calibrates a model by recursively averaging neighbouring non-monotonic scores, so that a piece-wise constant non-decreasing calibration map is obtained at the end. The main issue with isotonic calibration is that it is prone to overfitting on smaller datasets.

Beta calibration [16] is a recently proposed parametric approach for calibration of probabilistic two-class classifiers. The method has been derived from the assumption that among all instances of any one of the classes, the predicted probability to belong to class 1 is distributed according to a Beta distribution. Then, the calibration map as a transformation of the predicted probability to belong to class 1, can be shown to have the following parametric form:

$$\mathsf{c}_{\beta}(\mathbf{s}) = \Phi\Big(a \ln s_1 - b \ln s_2 + c\Big) \,, \tag{2}$$

where $\Phi(z) = (1 + e^{-z})^{-1}$ is the logistic sigmoid function, a, b and c are parameters depending on the Beta parameters, as well as the marginal class distribution $\mathsf{P}(Y)$. Beta calibration has the advantage of being naturally defined as a valid calibration map on the interval $[0, 1]$, while providing flexible shapes including sigmoids, inverse-sigmoids, and the identity map (*i.e.* no transformation is applied), as shown in Fig. 2 (right). Experimentally, it has been shown to provide good calibration results on various binary models such as AdaBoost, Support Vector Machines, and NNs [16, 17].

2.2 Quantile-Calibrated Regression

In regression, calibration has been traditionally addressed through quantiles [11, 26, 32, 3, 14]. The goal of a quantile regression model is for a given instance with feature vector $\mathbf{x}$ and for a given quantile $\tau \in [0, 1]$, to find an estimate $\hat{y}$ such that $\mathsf{P}(Y_\tau \leq \hat{y} \mid X = \mathbf{x}) = \tau$. The definition of calibrated quantile regression can then be given as:

Definition 2 (Quantile-Calibrated Regressor)**.** *Suppose we have a pair of jointly distributed random variables* (X, Y) *over* $\mathbb{X}$ *and* $\mathbb{Y} \subseteq \mathbb{R}$*, and a quantile regression model* $\mathsf{g} : \mathbb{X} \times [0, 1] \to \mathbb{Y}$*. Denoting* $G_\tau = \mathsf{g}(X, \tau)$ *as the random variable of the* τ*-quantile predictions,* g *is said to be quantile-calibrated iff,* $\forall \tau \in [0, 1]$*, the following holds:*

$$\mathsf{P}(Y \leq G_\tau) = \tau \,. \tag{3}$$

Motivated by the above definition, [14] introduces a quantile calibration map, which maps any quantile $\tau \in [0, 1]$ into $\mathsf{c}(\tau) = \mathsf{P}(Y \leq G_\tau) \in [0, 1]$. After applying this calibration map, the obtained quantile regression model is indeed quantile-calibrated, as $\mathsf{g}_{\mathsf{cal}}(X, \mathsf{c}(\tau)) = \mathsf{g}(X, \tau)$ and therefore, $\mathsf{P}(Y \leq \mathsf{g}_{\mathsf{cal}}(X, \mathsf{c}(\tau))) = \mathsf{P}(Y \leq \mathsf{g}(X, \tau)) = \mathsf{c}(\tau)$. To learn the mapping c, [14] use isotonic regression in the following way. Denoting by τ_i the quantile which corresponds to the actual target value y_i of the training instance $\mathbf{x}_i$, the proposed method starts by collecting the empirical frequency $\bar{\tau}_i = n^{-1} \sum_{j=1}^{n} \mathsf{I}(\tau_j \leq \tau_i)$ (where $\mathsf{I}(\cdot)$ is the indicator function), that quantifies the proportion of instances receiving a CDF value no greater than each given τ_i. Isotonic regression can then be applied to learn a mapping using the gathered pairs $(\tau_i, \bar{\tau}_i)_{i=1}^{n}$ to provide better calibrated quantiles according to the collected empirical frequency.

Remark 1 (Global vs Local Calibration)**.** Comparing Def. 1 with Def. 2, we see that classification is defined through a conditional probability, while for quantile regression it is only through a marginal probability. If a calibrated classifier provides a prediction with probability vector $\mathbf{s}$, then on average over all cases with the same prediction, the corresponding targets are distributed according to $\mathbf{s}$. In contrast, if a quantile-calibrated regression model provides a prediction with some empirical moments (*e.g.* mean and variance), then we cannot claim that on average over all cases with the

same predicted moments, the target variable would indeed have a distribution with the same true moments. This is because a marginal probability only considers the problem on a global level, which only guarantees the quantile to be calibrated averaged over all predictions. This property becomes a disadvantage when the goal is to quantify the uncertainties on each individual prediction. To mirror more closely the classification case, we next propose a stronger definition of calibration for regression.

3 Distribution-Calibrated Regression

The following definition originates from the same principle as Def. 1 for classification, in the sense that the distribution of the target variable Y is required to agree with the output of the model conditioned on the output of the model. We first choose $\mathbb{S}_{\mathbb{Y}}$ to consist of all possible probability distribution functions (PDFs) over a real-valued target variable. Note that by this we are restricting ourselves to working with absolutely continuous distributions.

Definition 3 (Distribution-Calibrated Regressor)**.** *Suppose we have a pair of jointly distributed random variables (X, Y) over $\mathbb{X}$ and $\mathbb{Y} \subseteq \mathbb{R}$, and a model $\mathsf{f} : \mathbb{X} \to \mathbb{S}_{\mathbb{Y}}$. Denoting $S = \mathsf{f}(X)$ as the random variable of model predictions, f is said to be distribution-calibrated if and only if $\forall \mathsf{s} \in \mathbb{S}_{\mathbb{Y}}$, $\forall y \in \mathbb{Y}$, the following equality holds:*

$$\mathsf{p}(Y = y \mid S = \mathsf{s}) = \mathsf{s}(y)\,. \tag{4}$$

In particular, this definition implies that if a calibrated model predicts a distribution with some mean μ and variance σ^2, then it means that on average over all cases with the same prediction the mean of the target is μ and variance is σ^2.

Next, we show that if the probabilistic regression model is distribution-calibrated then for any $\tau \in [0, 1]$ extracting the τ-quantile from the output distribution results in a quantile-calibrated regressor.

Theorem 1. *Let $\mathsf{f} : \mathbb{X} \to \mathbb{S}_{\mathbb{Y}}$ be a distribution-calibrated probabilistic model, and let $\mathsf{g} : \mathbb{X} \times [0, 1] \to \mathbb{Y}$ be a quantile regressor defined by $\mathsf{g}(\mathbf{x}, \tau) = y$ such that $\mathsf{P}_{\mathsf{f}(\mathbf{x})}(Y \leq y) = \tau$ where $\mathsf{P}_{\mathsf{f}(\mathbf{x})}$ is the probability measure corresponding to the distribution $\mathsf{f}(\mathbf{x})$. Then g is quantile-calibrated.*

Proof. First let us prove that g exists. Since $\mathsf{f}(\mathbf{x})$ is absolutely continuous then its CDF is continuous. As it is monotonic in the range $[0, 1]$ it achieves all values, including τ. Therefore, the required y exists for any $\mathbf{x}$ and τ and so g is correctly defined. It remains to prove that $\mathsf{P}(Y \leq \mathsf{g}(X, \tau)) = \tau$ for any $\tau \in [0, 1]$. As f is distribution-calibrated then for any $\mathsf{s} \in \mathbb{S}_{\mathbb{Y}}$ we get $\mathsf{P}(Y = y \mid \mathsf{f}(X) = \mathsf{s}) = \mathsf{s}(y)$ for any y. Combining this over all $y \leq \mathsf{g}(X, \tau)$ and considering that $\mathsf{g}(X, \tau)$ is uniquely determined by $\mathsf{f}(X)$, we obtain $\mathsf{P}(Y \leq \mathsf{g}(X, \tau) \mid \mathsf{f}(X) = \mathsf{s}) = \mathsf{P}_{\mathsf{f}(X)}(Y \leq \mathsf{g}(X, \tau))$ which is equal to τ due to the definition of g. Since $\mathsf{P}(Y \leq \mathsf{g}(X, \tau) \mid \mathsf{f}(X) = \mathsf{s}) = \tau$ for any s then $\mathsf{P}\Big(Y \leq \mathsf{g}(X, \tau)\Big) = \tau$ implying that g is quantile-calibrated. □

On the other hand, since $\mathsf{P}(Y \leq G_\tau) = \tau$ doesn't provide any information about $\mathsf{P}(Y \leq G_\tau \mid \mathsf{f}(X) = \mathsf{s})$, we don't necessarily get a distribution-calibrated model from a quantile calibrated model.

While we have provided the definition of distribution calibration and shown its relationship to quantile calibration, we now demonstrate the quantitative benefits for any model being calibrated on a distribution level. The metric we adopt is the negative log likelihood (NLL), also known as the log-loss in the context of proper scoring rules [6].

As shown in [15], the expected NLL can be decomposed into the following two terms:

$$\mathsf{E}_{(\mathbf{X},Y)}\Big[-\ln \mathsf{s}_{\mathbf{x}}(y)\Big] = \tag{5}$$
$$\mathsf{E}_{\mathbf{X}}\Big[\mathsf{KL}\Big(\mathsf{p}(Y \mid \mathsf{s}_{\mathbf{x}}), \mathsf{s}_{\mathbf{x}}(Y)\Big)\Big] + \mathsf{E}_{(\mathbf{X},Y)}\Big[-\ln \mathsf{p}(y \mid \mathsf{s}_{\mathbf{x}})\Big]\,,$$

where

$$\mathsf{KL}\Big(\mathsf{p}(Y \mid \mathsf{s}_{\mathbf{x}}), \mathsf{s}_{\mathbf{x}}(Y)\Big) = \int_y \mathsf{p}(Y \mid \mathsf{s}_{\mathbf{x}}) \ln \frac{\mathsf{p}(Y \mid \mathsf{s}_{\mathbf{x}})}{\mathsf{s}_{\mathbf{x}}(y)} \mathsf{d}\, y\,.$$

The first term is commonly known as the calibration loss and the latter term is the so called refinement loss. Once a model is trained, the latter term becomes fixed, as the distribution over S is learnt. Therefore, the whole expectation can be minimised if and only if the formal KL divergence becomes 0 everywhere, which indicates calibrated distributions as we defined. In general, distribution-level calibration ensures we have the most accurate uncertainty from the model predictions of the targets receiving the same prediction. Such calibration properties allow us to make optimal decisions for each individual prediction.

4 Methodology

The proposed idea is to use Beta calibration maps to transform CDFs of the distributions output by the regressor, similarly as isotonic maps are used by [14]. Unlike [14], we learn a GP to predict the parameters a, b, c of the Beta calibration map from the mean and variance as predicted by the regressor. Let us now look into these steps in more detail.

4.1 Beta link function for regression

We first adopt the parametric Beta calibration map family [16, 17] as a tool to calibrate the CDF of any regression output, by transforming quantiles with a beta calibration

map $[0,1] \to [0,1]$. As the Beta family contains the identity map ($a = 1, b = 1, c = 0$), the regression output can remain the same, if already calibrated. Changing c pushes the distribution to the left ($c > 0$) or right ($c < 0$). Sigmoids ($a, b > 1$) decrease the variance of the regression output distribution, while inverse sigmoids ($a, b < 1$) increase the variance. Changing the balance between a and b makes the distribution skewed to the left ($a < b$) or right ($a > b$).

Beta calibration map applies to the CDFs, while in order to later learn a GP we will need to know the transformation as a link function which directly applies to the PDFs. To derive this Beta link function, we need to differentiate the new CDF obtained after applying Beta calibration map c_β. Denoting the quantile by q_y, the differentiation results in the following:

$$\frac{\mathsf{d}\, \mathsf{c}_\beta\left(\mathsf{q}_y\right)}{\mathsf{d}\, y} = \frac{\mathsf{d}\, \mathsf{c}_\beta\left(\mathsf{q}_y\right)}{\mathsf{d}\, \mathsf{q}_y} \frac{\mathsf{d}\, \mathsf{q}_y}{\mathsf{d}\, y} = \mathsf{r}_\beta\left(\mathsf{q}_y\right) \mathsf{s}_y \,, \tag{6}$$

where $\mathsf{r}_\beta(\mathsf{q})$ is the link function that we were looking for:

$$\begin{aligned} \mathsf{r}_\beta(\mathsf{q}_y) &= \frac{\mathsf{d}\, \Phi\Big(a \ln \mathsf{q}_y - b \ln\left(1 - \mathsf{q}_y\right) + c\Big)}{\mathsf{d}\, \mathsf{q}_y} \\ &= \frac{\mathsf{q}_y^a (1 - \mathsf{q}_y)^b e^{-c} \left(a - (a-b)\mathsf{q}_y\right)}{\mathsf{q}_y(1-\mathsf{q}_y)\left(\mathsf{q}^a + (1-\mathsf{q}_y)^b e^{-c}\right)^2} \,. \end{aligned} \tag{7}$$

Here a, b and c are the parameters of Beta calibration. This link function acts as a density ratio between the calibrated PDF and the original PDF as output by the regressor.

A sufficient and necessary condition for the ratio to be non-negative is $a \geq 0$ and $b \geq 0$, which is the same condition as required to have a monotonically increasing Beta calibration map c_β. Similarly, the parameter setting $a = 1$, $b = 1$ and $c = 0$ gives a constant ratio of 1 (*e.g.* no adjustment on the PDF), corresponding to the identify calibration map of c_β. Furthermore, as it is defined over the CDF, r_β has the advantage of always yielding normalised distribution after the multiplication, which can otherwise only be achieved through constrained optimisation for common models in the field of density ratio estimation [31].

4.2 The GP-BETA Model

With r_β being defined as above, intuitively, the next step to achieve distribution calibration is to construct a model that maps any regression output (μ_i, σ_i) into a set of Beta calibration parameters (a_i, b_i, c_i). However, the main challenge of this approach is that, during training, for each (μ_i, σ_i) we normally only observe a single target value y_i, which is not enough to learn a good calibration map on the whole distribution. Therefore, we seek to borrow the observed values from other regression outputs that are close to (μ_i, σ_i), which leads to the choice of the GP, a widely adopted non-parametric method within the Bayesian framework.

The proposed model can be formalised in the following way. We first assume, that there are three latent functions that are jointly distributed with respect to a multi-output GP [1, 27, 19], corresponding to the parameters a, b, c of Beta calibration:

$$(\mathsf{w}_a, \mathsf{w}_b, \mathsf{w}_c) \sim \mathsf{gp}(0, \mathsf{k}, \mathbf{B}) \,, \tag{8}$$

where k is the kernel (covariance) function on the regression output distributions, and $\mathbf{B}$ is a 3×3 coregionalisation matrix modelling the covariance among the outputs.

Regarding the choice of k, here we refer to the well developed area of kernel mean embedding [20]. The idea is to use a kernel function to map each distribution into a reproducing kernel Hilbert space (RKHS), the embedding can then be applied to problems like the two sample test [8] and distribution regression [18]. Notice that in distribution regression the task is to predict a target variable from a distribution variable, in our model the task is further generalised to infer a distribution over functions from a distribution variable, for the purpose of calibration.

Calculating the embedding/kernel value generally requires Monte-Carlo samples from the candidate distributions, but can also be analytic under certain combinations of distributions. Here we choose the univariate Gaussian embedding with Radial Basis Function (RBF) kernel [29]:

$$\mathsf{k}\Big((\mu_1, \sigma_1), (\mu_2, \sigma_2)\Big) = \frac{\theta}{|\sigma_1 + \sigma_2 + \theta^2|^{\frac{1}{2}}} e^{\left(-\frac{(\mu_1 - \mu_2)^2}{2(\sigma_1 + \sigma_2 + \theta^2)}\right)} \,. \tag{9}$$

Observe that if we set $\sigma_1 = \sigma_2 = 0$, the kernel reduces to a RBF kernel defined over (μ_1, μ_2).

Given n training points $(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \Big((\mu_1, \sigma_1), \ldots, (\mu_n, \sigma_n)\Big)$, a Gaussian likelihood on $(w_a^{(i)}, w_b^{(i)}, w_c^{(i)})_{i=1}^n$ can then be written as:

$$\mathsf{p}\left(\begin{bmatrix}\mathbf{w}_a \\ \mathbf{w}_b \\ \mathbf{w}_c\end{bmatrix} \,\middle|\, \boldsymbol{\mu}, \boldsymbol{\sigma}\right) = \mathsf{N}\left(\begin{bmatrix}\mathbf{w}_a \\ \mathbf{w}_b \\ \mathbf{w}_c\end{bmatrix} \,\middle|\, \mathbf{0}, \mathbf{B} \otimes \mathbf{K}\right) \,, \tag{10}$$

where N is the likelihood function of multivariate Gaussian, $\mathbf{K}$ is the n by n kernel matrix obtained by applying k on $(\boldsymbol{\mu}, \boldsymbol{\sigma})$. $\mathbf{B}$ is the coregionalisation matrix introduced above, while $\otimes$ denotes the Kronecker product. $\mathbf{w}_a = [w_a^{(1)}, \ldots, w_a^{(n)}]$ and similar for $\mathbf{w}_b$, $\mathbf{w}_c$.

While the form above gives a clear representation of the coregionalisation structure, we use $\mathbf{C} = \mathbf{K} \otimes \mathbf{B}$ for the rest

of the paper for convenience,

$$\mathsf{p}(\mathbf{w} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) = \mathsf{N}(\mathbf{w} \mid \mathbf{0}, \mathbf{C}) \,,$$
$$\mathbf{w} = \left[\mathbf{w}_1^\mathsf{T}, \ldots, \mathbf{w}_m^\mathsf{T}\right]^\mathsf{T}, \quad \mathbf{w}_i = [w_a^{(i)}, w_b^{(i)}, w_c^{(i)}]^\mathsf{T} \,.$$

For target values $\mathbf{y} = (y_1, \ldots, y_n)$, we can now plug in our Beta link:

$$\mathsf{p}(\mathbf{y} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) = \int_{\mathbf{w}} \Big(\prod_{i=1}^{n} \mathsf{p}(y_i \mid \mathbf{w}_i, \mu_i, \sigma_i) \Big) \, \mathsf{p}(\mathbf{w} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) \, d\mathbf{w} \,,$$
$$\mathsf{p}(y_i \mid \mathbf{w}_i, \mu_i, \sigma_i) = \mathsf{s}_{y_i} \, \mathsf{r}_\beta^{(i)}(\mathsf{q}_{y_i}) \,,$$

where q_{y_i} and s_{y_i} represent the Gaussian PDF value and CDF value at y_i given μ_i and σ_i, $\mathsf{r}_\beta^{(i)}$ is the link with parameters a_i, b_i, and c_i given as:

$$a_i = e^{(\gamma_a^{-1} \, w_a^{(i)} + \delta_a)} \,,$$
$$b_i = e^{(\gamma_b^{-1} \, w_b^{(i)} + \delta_b)} \,,$$
$$c_i = \gamma_c^{-1} \, w_c^{(i)} + \delta_c \,.$$

The exponential function enforces the non-negative constraints on a and b. The hyperparameters $(\gamma., \delta.)$ control the link function at the $\mathbf{0}$-mean GP prior. For distribution calibration, a reasonable prior is to use the identity calibration map, indicating that we should not adjust the density functions before seeing any data. However, while at the prior we have $\left(-e^{\mathsf{E}(w_a^{(i)})} = 1, e^{\mathsf{E}(w_b^{(i)})} = 1, \mathsf{E}(w_c^{(i)}) = 0\right)$ corresponding to the identity map, the Gaussian variance together with the non-linear transform will distort the calibration map after marginalising $\mathbf{w}_i$. While this distortion cannot be prevented analytically, we use the hyperparameters above to reduce the level of distortion, and optimise them during training time, in the spirit of empirical Bayes methods [25].

4.3 Scalable Inference

We have defined $\mathsf{p}(\mathbf{y} \mid \boldsymbol{\mu}, \boldsymbol{\sigma})$ in Eq. (11); however the integral is analytically intractable due to the non-linearity in the link function, which makes optimising the hyperparameters challenging. Furthermore, given a test instance $(\mu_\star, \sigma_\star)$, the calibrated density value at $\mathsf{s}_\star(y)$ are also intractable:

$$\hat{\mathsf{s}}_\star(y) = \int_{\mathbf{w}_\star} \int_{\mathbf{w}} \mathsf{r}_\beta^{(\star)} \mathsf{s}_\star(y) \mathsf{p}(\mathbf{w}_\star \mid \mathbf{w}) \mathsf{p}(\mathbf{w} \mid \mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}) d\mathbf{w} d\mathbf{w}_\star \,,$$

$$\mathsf{p}(\mathbf{w} \mid \mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \propto \Big(\prod_{i=1}^{n} \mathsf{p}(y_i \mid \mathbf{w}_i, \mu_i, \sigma_i) \Big) \, \mathsf{p}(\mathbf{w} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) \,,$$

Finally, operations on the kernel matrix also present computational challenges as the number of data points grows. These issues are analogous to the ones seen in GP classification, which also needs to integrate over a Gaussian likelihood to get a non-Gaussian likelihood, as well as dealing with computations on the kernel matrix.

We therefore introduce the scalable inference scheme as proposed in [10], together with the Monte-Carlo gradients approach to address the intractable integration on the link function. The inference scheme starts with placing a number of m induced pseudo points as in [28, 21], denoting as $(\boldsymbol{\mu}_\mathbf{u}, \boldsymbol{\sigma}_\mathbf{u})$, from which we obtain a Gaussian prior $\mathsf{N}(\mathbf{u} \mid \mathbf{0}, \mathbf{C}_u)$, with $\mathbf{C}_u$ being a $3m$ by $3m$ covariance matrix obtained from the kernel function and coregionalisation matrix. The task is then to approximate a Gaussian posterior $\mathsf{q}(\mathbf{u} \mid \mathbf{m}_\mathbf{u}, \mathbf{V}_\mathbf{u})$ with the parameters $\mathbf{m}_\mathbf{u}$ and $\mathbf{V}_\mathbf{u}$, the evidence lower bound as seen in common variational inference approaches:

$$\mathsf{p}(\mathbf{y}) \geq \mathsf{E}_{\mathsf{q}(\mathbf{u})}[\ln \mathsf{p}(\mathbf{y} \mid \mathbf{u})] - \mathsf{KL}[\mathsf{q}(\mathbf{u}), \mathsf{N}(\mathbf{u})] \,, \tag{11}$$

where we omit the dependence on the inputs for simplicity. While the KL-divergence can be computed analytically, the expectation $\mathsf{E}_{\mathsf{q}(\mathbf{u})}[\ln \mathsf{p}(\mathbf{y} \mid \mathbf{u})]$ still remains intractable, as it requires the computation of $\ln \int_\mathbf{w} \mathsf{p}(\mathbf{y} \mid \mathbf{w}) \mathsf{p}(\mathbf{w} \mid \mathbf{u}) \mathrm{d}\mathbf{w}$. As a solution, in [10] the authors propose to apply the Jensen inequality again to obtain a further bound:

$$\begin{aligned}
&\mathsf{E}_{\mathsf{q}(\mathbf{u})}[\ln \mathsf{p}(\mathbf{y} \mid \mathbf{u})] - \mathsf{KL}[\mathsf{q}(\mathbf{u}), \mathsf{N}(\mathbf{u})] \\
&\quad \geq \mathbf{E}_{\mathsf{q}(\mathbf{u})}\left[\mathsf{E}_{\mathsf{p}(\mathbf{w}|\mathbf{u})}[\ln \mathsf{p}(\mathbf{y} \mid \mathbf{w})]\right] - \mathsf{KL}[\mathsf{q}(\mathbf{u}), \mathsf{N}(\mathbf{u})] \,, \\
&\quad = \mathsf{E}_{\mathsf{q}(\mathbf{w})}[\ln \mathsf{p}(\mathbf{y} \mid \mathbf{w})] - \mathsf{KL}[\mathsf{q}(\mathbf{u}), \mathsf{N}(\mathbf{u})] \,,
\end{aligned}$$

where

$$\begin{aligned}
\mathsf{q}(\mathbf{w}) &= \mathsf{N}(\mathbf{w} \mid \mathbf{m}_\mathbf{w}, \mathbf{V}_\mathbf{w}) \,, \\
\mathbf{m}_\mathbf{w} &= \mathbf{A}\mathbf{m}_\mathbf{u} \,, \\
\mathbf{V}_\mathbf{w} &= \mathbf{C} + \mathbf{A}(\mathbf{V}_\mathbf{u} - \mathbf{C}_\mathbf{u})\mathbf{A}^\mathsf{T} \,, \\
\mathbf{A} &= \mathbf{C}_\mathbf{wu}\mathbf{C}_\mathbf{u}^{-1} \,,
\end{aligned} \tag{12}$$

where $\mathbf{C}$, $\mathbf{C}_\mathbf{u}$ are as before, $\mathbf{C}_\mathbf{wu}$ is the $3n$ by $3m$ kernel matrix between the training points and the inducing points.

We can now compute the the expectation $\mathsf{E}_{\mathsf{q}(\mathbf{w})}[\ln \mathsf{p}(\mathbf{y} \mid \mathbf{w})]$ via Monte-Carlo samples. In fact, each $\ln \mathsf{p}(y_i \mid \mathbf{w}_i)$ can be efficiently computed via three-dimensional Gaussian samples by selecting the corresponding $\mathbf{m}_{\mathbf{w}_i}$ and $\mathbf{V}_{\mathbf{w}_i}$, and performing a reparameterization trick:

$$\boldsymbol{\epsilon}_{\mathbf{w}_i} = \mathbf{L}_{\mathbf{w}_i}\boldsymbol{\epsilon} + \mathbf{m}_{\mathbf{w}_i} \,,$$
$$\mathbf{L}_{\mathbf{w}_i} = \mathsf{Cholesky}(\mathbf{V}_{\mathbf{w}_i}) \,,$$

with $\boldsymbol{\epsilon}$ being random samples generated from a three dimensional unit Gaussian. Such a reparameterization allows us to compute the gradient over $\mathbf{m}_{\mathbf{w}_i}$, $\mathbf{V}_{\mathbf{w}_i}$ through the Monte-Carlo integration, which can then be used to opti-

mise all the parameters and hyperparameters. In general, for the whole model we have the following parameters to optimise: the kernel parameters $(\theta, \mathbf{B})$, the varational parameters $(\mathbf{m_u}, \mathbf{V_u})$, link parameters (γ, δ), and the locations for the inducing points $(\boldsymbol{\mu_u}, \boldsymbol{\sigma_u})$. Additionally, as suggested by [10], we replace the parameter $\mathbf{V_u}$ by its Cholesky factor to ensure $\mathbf{V_u}$ to be always positive definite. The overall model can be computed efficiently using modern frameworks supporting automatic differentiation, and can be easily scaled via both online and distributed training using stochastic gradient descent. To predict a new test instance, the procedure duplicates the one given in Eq. (12): we first compute mean and covariance of $\mathbf{w}_\star$ at the test point, then compute the calibrated densities through Monte-Carlo integration.

The overall computational cost includes the cost of calculating the KL-bound (same as in [10]) and the cost of Monte-Carlo integration within gradient calculation. During training time, for m inducing points, computing the KL bound requires $\mathcal{O}(m^3)$. Computing the link function and its gradient on a batch takes $\mathcal{O}(n * l)$, where n is the size of the mini-batch, and l is the number of Monte-Carlo samples. At prediction time, a single instance will cost $\mathcal{O}(m^2 + l)$. Both l and m can be selected by the user, so the overall time cost is manageable on personal computers and can be significantly shortened using common deep learning frameworks with GPU acceleration.

5 Experiments

We now provide emperical analysis of the GP-BETA method in the context of distribution calibration.

Synthetic data. As shown in Fig. 1, we generate the synthetic dataset of 360 points via an equal mixture of two univariate linear models: $y = 0.5x + \epsilon$; and $y = \epsilon$, with $\epsilon \sim \mathsf{N}(0, \sqrt{2})$, uniformly sampled within $[-10, 40]$. The applied OLS tends to fit a line in the centre as it can only model a uni-modal Gaussian conditional density.

We apply both isotonic regression and GP-BETA to examine their behaviour under such a scenario. The CDF is almost unaffected after applying the isotonic approach, due to the fact that the original OLS was close to being (marginally) quantile calibrated. Careful inspection shows some fluctuating patterns in the PDFs, caused by the step-wise nature of isotonic regression. Isotonic calibration leads to non-smooth PDFs after calibrating the quantiles.

Finally, we analyse the results obtained from a GP-BETA model with 32 inducing points. The model is trained using the ADAM optimiser with a learning rate of 0.01. The resulting PDF is able to capture the high density region around the bottom left region, where the two original linear models overlap. Towards the right side, the GP-BETA model is able to recover the bi-modal nature from the true conditional density, only having access to the Gaussian distribution predicted by the OLS. On the CDF, we can observe that the GP-BETA model is able to adjust CDF on different scales conditioning on the original model output, which gives better recovery of the true CDF. This result can be further illustrated through Fig. 2. As the figure indicates, the GP-BETA model provides different estimations for each given output. Corresponding calibration maps can hence be obtained via Monte-Carlo samples as mentioned previously, from which we show three examples on the right of Fig. 2. This helps us to further demonstrate the purpose of distribution calibration. As the isotonic approach only aims for calibrating the quantiles, it uses the same calibration map on each model output, and provides limited improvements for quantifying model uncertainty in the case of our synthetic dataset. GP-BETA, on the other hand, is designed to work towards calibrated distributions, which by definition requires conditional calibration maps.

Real world datasets. We focus on three evaluation measures: (i) predictive negative log likelihood (NLL), (ii) mean squared error (MSE), and (iii) pinball loss (PBL). As discussed previously, for a pre-trained model, the NLL will be minimised if a model achieves density-level calibration. MSE, on the other hand, is a generic measure to evaluate a model's predictive performance. Pinball loss commonly used to train and evaluate the calibration of quantiles [3], and is defined as follows:

$$\mathsf{PBL}(\tau) = \mathsf{E}_{(\mathbf{X},Y)}[\mathsf{L}(y, g(\mathbf{x}, \tau))]\,,$$

$$\mathsf{L}\Big(y, g(\mathbf{x}, \tau)\Big) = \begin{cases} (1-\tau)\Big(g(\mathbf{x}, \tau) - y\Big) & \text{if } y < g(\mathbf{x}, \tau)\,, \\ \tau\Big(y - g(\mathbf{x}, \tau)\Big) & \text{otherwise}\,. \end{cases}$$

Pinball loss is an asymmetric loss where the overestimation loss and underestimation loss are weighted with the predicted quantiles, and hence specified for each given quantile. In the following experiments we calculate the averaged loss from the quantiles of 0.05 to 0.95, in increments of 0.05.

We select the following four regression models: 1. OLS regression, 2. Bayesian Ridge Regression (BRR), 3. GP Regression (GPR), and 4. BNNs.

The first two models provide uniform variance estimates for each instance; BRR optimises the variance using an inverse-gamma prior. The later two provide variance estimates for each instance. While GPR is derived within the non-parametric Bayesian framework, the NN model doesn't provide uncertainty estimations by default, but through the use of dropout approximations, we can obtain some form of uncertainty around the observations [5].

The experiments are applied on the following UCI datasets (sizes in parentheses): 1. Diabetes (442), 2. Boston (506), 3. Airfoil (1503), 4. Forest Fire (517), 5. Strength (1030),

OLS

Dataset	NLL						MSE						PBL					
	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}
1	5.37	5.78	**5.34**	5.46	5.39	5.38	2664.69	2664.86	**2590.12**	2603.50	2616.56	2640.62	1720.24	1720.84	**1684.4**	1760.26	1715.	1710.14
2	3.04	3.2	2.84	2.85	2.85	**2.83**	25.28	25.31	22.03	21.20	22.23	**20.75**	178.34	176.65	160.37	157.32	159.79	**155.17**
3	2.99	3.14	2.93	**2.92**	2.92	2.92	23.00	22.98	21.78	21.45	21.37	**21.28**	524.63	523.99	506.	503.39	502.78	**501.**
4	1.94	2.22	1.84	**1.81**	3.29	2.25	2.56	2.57	2.85	**2.45**	3.2	3.83	59.94	**59.28**	64.40	60.25	69.12	76.41
5	3.76	4.27	3.76	3.73	3.73	**3.71**	108.77	108.55	108.89	107.84	106.08	**105.24**	788.66	789.96	789.37	779.02	771.04	**767.79**
6	5.94	**5.37**	5.56	5.47	5.44	5.47	**8502.15**	8502.19	8624.57	8528.08	8556.49	8528.08	112557.73	**98345.03**	100867.49	100962.09	100708.98	100962.09

BR

Dataset	NLL						MSE						PBL					
	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}
1	5.33	5.83	5.51	5.55	5.5	**5.28**	2398.89	2404.58	2421.27	**2333.93**	2405.24	2421.43	1609.63	1620.15	1738.29	1758.86	1728.5	**1596.14**
2	2.94	3.06	2.74	2.73	2.73	**2.73**	20.65	20.53	**15.45**	15.96	16.12	15.97	169.16	164.01	**142.45**	143.54	142.63	142.58
3	2.96	3.15	2.91	2.92	2.89	**2.89**	21.65	21.63	19.54	19.62	**19.44**	19.46	513.84	512.89	487.04	488.23	483.71	**482.61**
4	1.83	2.31	2.16	**1.80**	1.85	1.81	**2.22**	2.22	2.75	2.57	2.65	2.57	55.90	**53.22**	59.86	57.74	58.88	57.98
5	3.76	4.01	3.75	3.73	**3.73**	3.74	107.80	107.82	108.41	**107.78**	107.82	108.44	791.3	791.23	786.36	782.1	**779.74**	784.40
6	5.96	**5.4**	5.54	5.49	5.41	5.44	8750.67	**8750.46**	9162.78	9382.19	8800.73	8771.09	113298.92	**98739.99**	101289.49	106380.51	98973.39	99643.87

NN

Dataset	NLL						MSE						PBL					
	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}
1	7.30	6.08	5.43	5.44	**5.42**	5.44	3584.02	3550.86	3521.95	3542.58	3523.05	**3517.75**	2303.88	2004.76	1969.62	1978.68	**1966.**	1975.43
2	2.78	2.86	2.74	2.75	2.72	**2.72**	15.85	15.89	15.16	14.66	**14.51**	14.69	145.90	144.40	141.61	141.01	**138.32**	139.1
3	5.96	5.04	3.78	3.72	3.77	**3.67**	1031.88	1367.93	64.48	**61.53**	63.14	62.80	4044.07	3867.07	1018.82	976.44	1003.24	**945.84**
4	7.49	16.71	1.68	1.68	1.64	**1.58**	2.36	2.35	**2.3**	2.33	2.38	2.35	68.86	57.64	56.28	56.52	56.47	**56.14**
5	3.29	3.45	3.16	3.16	**3.16**	3.16	53.59	**41.38**	42.69	42.57	42.90	42.79	529.13	**461.93**	464.56	465.01	466.24	465.81
6	18.09	7.10	5.49	5.25	**5.22**	5.25	**9833.01**	9840.09	11206.57	12011.97	9890.18	12011.97	110156.3	**98102.22**	117965.81	112022.83	98574.75	112022.83

GP

Dataset	NLL						MSE						PBL					
	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}	Base	ISO	GPB_8	GPB_{16}	GPB_{32}	GPB_{64}
1	5.43	5.74	**5.43**	5.43	5.43	5.43	3022.28	3022.22	**3017.1**	3030.25	3027.80	3017.9	1820.38	1822.93	**1818.42**	1821.11	1821.67	1819.82
2	2.59	2.77	2.46	2.43	2.44	**2.43**	9.86	9.85	9.08	8.66	8.86	**8.51**	116.33	113.23	108.35	105.36	106.25	**104.47**
3	3.13	3.33	3.28	3.19	3.25	**3.12**	30.13	30.08	29.79	**29.54**	29.55	29.54	407.47	407.28	430.65	415.09	425.39	**403.90**
4	1.84	**-0.41**	2.22	2.24	2.02	1.63	**2.27**	2.27	2.47	3.28	2.4	2.3	56.47	122.1	70.53	75.39	63.13	**54.51**
5	3.08	3.31	3.08	3.09	**3.08**	3.08	29.91	29.91	30.24	30.55	30.11	**29.91**	**394.32**	399.14	396.3	399.37	395.83	395.33
6	6.28	6.08	5.63	5.75	5.80	**5.46**	13307.05	13316.46	13548.90	13666.95	13739.56	**13284.93**	6343.86	5835.73	5788.3	5818.16	5789.79	**5762.03**

Table 1. Predictive performance with the different base models, the lowest loss is shown in bold.

6. Energy (19735). The exception is GPR, where we limit the maximal dataset size to 1000 data points, due to the computational complexity of GPs. All the experiments use a random $(0.75, 0.25)$ train-test split, with both the base model and calibrators trained on the same set. During prediction time, 4096 points with equal distance are selected from $\mu_{min} - 8\,\sigma_{max}$ to $\mu_{max} + 8\,\sigma_{max}$, where μ_{min}, μ_{max} are the minimal and maximal predicted mean values in the training set, and σ_{max} is the maximal of the predicted standard deviation. This ensures that we cover nearly the whole predicted and calibrated densities, allowing the expected value to be approximated through the trapezoid rule.

For the NNs, we use the same setting as in [14], which is a 2-layer fully-connected structure with 128 hidden units per layer and ReLU activation. The dropout rate is set to 0.5, default weight decay of 10^{-4} and the length scale of 1.0 are used to approximate the mean and variance following the results given in [5].

We run GP-BETA with 8, 16, 32 and 64 inducing points, batch size of 128, and 64 Monte-Carlo samples per batch to compute the objective function and the gradient. The parameters are again optimised using ADAM with a learning rate of 0.001. The results are given in Tab. 1, showing that for most cases the GP-BETA model is capable of improving the results on all the three evaluation measures. GP-BETA didn't show improvements in some cases for dataset 3, which has about 20000 instances. Increasing the number of inducing points can be potentially beneficial in such cases according existing sparse GP literature. Another observation is on dataset 4 with GP as the base model, where the isotonic approach gives a significantly low NLL. This dataset has a target distribution where about half of the target values are 0, there is a chance that the isotonic regression happens to assign a step change exactly or very close to 0, which results in a very high log-likelihood.

In summary, the GP-BETA method is clearly superior in the synthetic example where local calibration is required, and in most real-world examples. We also note that for NLL, which we argue is the most faithful metric for distribution calibration, GP-BETA almost always performs best irrespective of the choice of the number of inducing points.

6 Conclusions

While both calibration of classifiers and quantile regressors have been studied broadly, we introduce the idea of distribution calibration. Models that are well calibrated on a distribution level provide improved uncertainty quantification on the target variable, as well being calibrated on the quantile level.

Although distribution calibration is applicable to any conditional density estimator, we focus on a regression setting given its popularity in predictive machine learning tasks. We propose the GP-BETA approach which combines multi-output GPs with Beta calibration from binary classification, and distribution regression. The sparse variational inference scheme allows the model to scale to large datasets and shows strong empirical performance. Directions for further work include non-parametric calibration maps, and generalising the model to other forms of density estimation.

Acknowledgements

This work was supported by the SPHERE Interdisciplinary Research Collaboration, funded by the UK Engineering and Physical Sciences Research Council under grant EP/K031910/1. MK was supported by the Estonian Research Council under grant PUT1458.

References

[1] Alvarez, M. and Lawrence, N. Sparse convolved Gaussian processes for multi-output regression. In *Proceedings of 21st Advances in Neural Information Processing Systems (NIPS 2008)*, 2008.

[2] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[3] Fasiolo, M., Goude, Y., Nedellec, R., and Wood, S. N. Fast calibrated additive quantile regression. *arXiv preprint arXiv:1707.03307*, 2017.

[4] Fawcett, T. and Niculescu-Mizil, A. PAV and the ROC convex hull. *Machine Learning*, 68(1):97–106, 2007.

[5] Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1050–1059, 2016.

[6] Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[7] Gneiting, T., Balabdaoui, F., and Raftery, A. E. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

[8] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13 (1):723–773, 2012.

[9] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017.

[10] Hensman, J., Matthews, A., and Ghahramani, Z. Scalable variational Gaussian process classification. *Journal of Machine Learning Research*, 38:351–360, 2015.

[11] Ho, Y. H. S. and Lee, S. M. S. Calibrated interpolated confidence intervals for population quantiles. *Biometrika*, 92(1):234–241, 2005. ISSN 00063444.

[12] Jatta, J. S. and Krishnan, K. K. An empirical assessment of a univariate time series for demand planning in a demand-driven supply chain. *International Journal of Business Forecasting and Marketing Intelligence*, 2 (3):269–290, 2016.

[13] Koenker, R. and Hallock, K. F. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

[14] Kuleshov, V., Fenner, N., and Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2796–2804, 2018.

[15] Kull, M. and Flach, P. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Machine Learning and Knowledge Discovery in Databases*, pp. 68–85. Springer International Publishing, 2015.

[16] Kull, M., Filho, T. S., and Flach, P. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pp. 623–631, 2017.

[17] Kull, M., Silva Filho, T. M., Flach, P., et al. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052–5080, 2017.

[18] Mitrovic, J., Sejdinovic, D., and Teh, Y. W. DR-ABC: approximate Bayesian computation with kernel-based distribution regression. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[19] Moreno-Muñoz, P., Artés, A., and Álvarez, M. Heterogeneous multi-output gaussian process prediction. In *Advances in Neural Information Processing Systems*, pp. 6711–6720, 2018.

[20] Muandet, K., Fukumizu, K., Sriperumbudur, B., Schölkopf, B., et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10(1-2):1–141, 2017.

[21] Naish-Guzman, A. and Holden, S. The generalized FITC approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1064, 2008.

[22] Orallo, J. H. Probabilistic reframing for cost-sensitive regression. In *ACM Transactions on Knowledge Discovery from Data*, volume 8, pp. 1–55. Association for Computing Machinery (ACM), 2014.

[23] Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3): 61–74, 1999.

[24] Rasmussen, C. E. and Williams, C. K. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[25] Robbins, H. An empirical bayes approach to statistics. In *Herbert Robbins Selected Papers*, pp. 41–47. Springer, 1985.

[26] Rueda, M., Martínez-Puertas, S., Martínez-Puertas, H., and Arcos, A. Calibration methods for estimating quantiles. *Metrika*, 66(3):355–371, Nov 2007. ISSN 1435-926X.

[27] Skolidis, G. and Sanguinetti, G. Bayesian multitask classification with gaussian process priors. *IEEE Transactions on Neural Networks*, 22(12), 2011.

[28] Snelson, E. and Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pp. 1257–1264, 2006.

[29] Song, L., Zhang, X., Smola, A., Gretton, A., and Schölkopf, B. Tailoring density estimation via reproducing kernel moment matching. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 992–999. ACM, 2008.

[30] Sugiyama, M., Takeuchi, I., Suzuki, T., Kanamori, T., Hachiya, H., and Okanohara, D. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 781–788, 2010.

[31] Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[32] Taillardat, M., Mestre, O., Zamo, M., and Naveau, P. Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Monthly Weather Review*, 144(6):2375–2393, 2016.

[33] Zadrozny, B. Reducing multiclass to binary by coupling probability estimates. In *Advances in neural information processing systems*, pp. 1041–1048, 2002.